

**This Page Is Inserted by IFW Operations
and is not a part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- **BLACK BORDERS**
- **TEXT CUT OFF AT TOP, BOTTOM OR SIDES**
- **FADED TEXT**
- **ILLEGIBLE TEXT**
- **SKEWED/SLANTED IMAGES**
- **COLORED PHOTOS**
- **BLACK OR VERY BLACK AND WHITE DARK PHOTOS**
- **GRAY SCALE DOCUMENTS**

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,
please do not report the images to the
Image Problem Mailbox.**

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 08-180018

(43)Date of publication of application : 12.07.1996

(51)Int.Cl.

G06F 15/00
G06F 13/00
G06F 13/10
G06F 15/16

(21)Application number : 06-323117

(71)Applicant : TOSHIBA CORP

(22)Date of filing : 26.12.1994

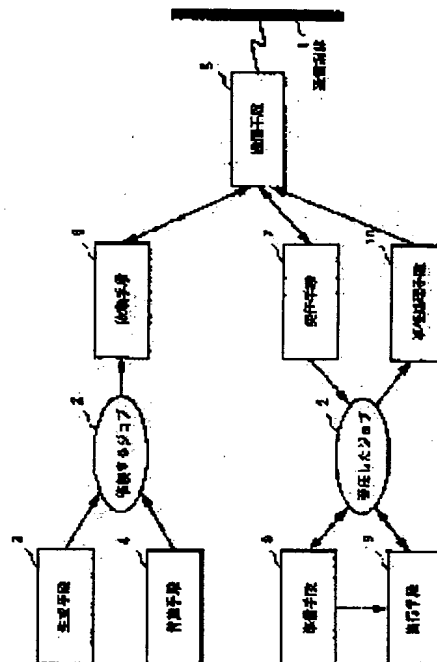
(72)Inventor : NUKUI HARUMI

(54) DISTRIBUTED PROCESSING SYSTEM AND DISTRIBUTED PROCESSING METHOD

(57)Abstract:

PURPOSE: To provide a distributed processing system and a distributed processing method whose constitution is simple.

CONSTITUTION: A generation means 3 generates a job 2 to be requested to another node, an addition means 4 adds auxiliary information for the execution of the procedure of the job 2 to the job 2 and a request means 6 requests the job 2 to the other node through a communication means 5 and a communication medium 1. In the other node, an accepting means 7 decides whether or not to accept the job. In the node which accepts it, a preparation means 8 prepares for the execution of the job 2 based on the auxiliary information. The preparation is the conversion of the procedure of a non-execution form to an execution form and the setting of a prescribed environment, etc., for instance. After the preparation, an execution means 9 executes the procedure (instruction) provided in the job 2 and a post-processing means 10 handles an executed result as instructed by the job 2 and typically returns it to a request destination (request origin).



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平8-180018

(43)公開日 平成8年(1996)7月12日

| (51)Int.Cl. ⁶ | 識別記号 | 庁内整理番号 | F I | 技術表示箇所 |
|--------------------------|---------|---------|-----|--------|
| G 0 6 F 15/00 | 3 1 0 H | 9364-5L | | |
| 13/00 | 3 5 5 | 7368-5E | | |
| 13/10 | 3 3 0 B | 7368-5E | | |
| 15/16 | 3 7 0 N | | | |

審査請求 未請求 請求項の数12 O L (全 16 頁)

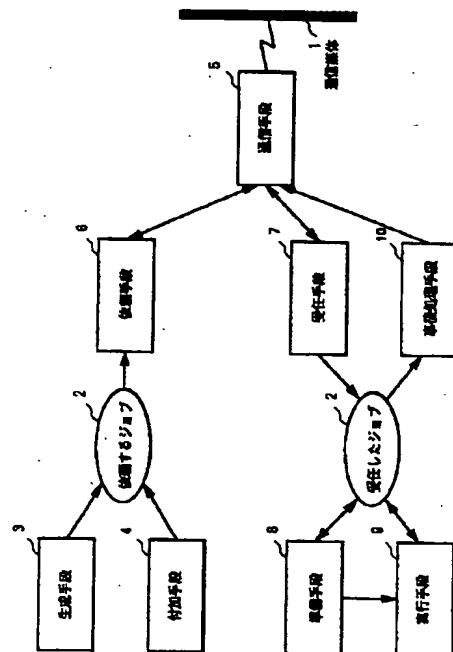
| | | | |
|----------|------------------|---------|--|
| (21)出願番号 | 特願平6-323117 | (71)出願人 | 000003078 株式会社東芝 神奈川県川崎市幸区堀川町72番地 |
| (22)出願日 | 平成6年(1994)12月26日 | (72)発明者 | 貫井 春美 神奈川県川崎市幸区柳町70番地 株式会社 東芝柳町工場内 |
| | | (74)代理人 | 弁理士 木内 光春 |

(54) 【発明の名称】 分散処理システム及び分散処理方法

(57) 【要約】

【目的】 構成が単純な分散処理システム及び分散処理方法を提供する。

【構成】 生成手段3が、他のノードに依頼すべきジョブ2を生成する。付加手段4が、ジョブ2の手続きの実行のための補助情報をジョブ2に付加する。依頼手段6あ、通信手段5及び通信媒体1を通じてジョブ2を他のノードに依頼する。他のノードでは、受任手段7がジョブを受任するか否か決定する。受任したノードでは、準備手段8が補助情報に基づいてジョブ2実行の準備を行う。この準備は、例えば非実行形態の手続きを実行形態に変換したり、所定の環境の設定などである。準備の後、実行手段9がジョブ2に含まれる手続き（命令）を実行する。事後処理手段10が、実行結果をジョブ2で指示されたように扱うが、典型的には要求先（依頼元）に返送する。



1

2

【特許請求の範囲】

【請求項1】 通信媒体によって接続された複数のノードを含む分散処理システムにおいて、各ノードが、前記通信媒体を通じて他のノードと通信するための通信手段と、他のノードに依頼すべきジョブを生成する生成手段と、前記ジョブに、ジョブの命令の実行のための補助情報を付加する付加手段と、前記ジョブを前記通信手段を通じて他のノードに送信して実行を依頼する依頼手段と、他のノードから送信されたジョブを受任する受任手段と、受任したジョブの前記補助情報に基づいて、当該ジョブの実行の準備を行う準備手段と、ジョブの命令を実行する実行手段と、前記実行の結果を処理する事後処理手段と、を有することを特徴とする分散処理システム。

【請求項2】 前記受任手段は、他のノードから送信されてきたジョブを受任するか否かを決定するように構成されたことを特徴とする請求項1記載の分散処理システム。

【請求項3】 前記依頼手段は、前記ジョブを全部又は一部の前記ノードに送信するように構成され、前記受任手段は、ジョブが送信されたノードにおいて、送信されたジョブを受任できると決定したときは、当該ジョブを送信した依頼側ノードに仮承諾を返信するように構成され、

前記依頼手段は、依頼側ノードにおいて、最早の前記仮承諾を返信したノードに正式依頼を送信することによってジョブの依頼を行うように構成されたことを特徴とする請求項2記載の分散処理システム。

【請求項4】 前記生成手段は、所定の場合に複数のジョブからなる複合ジョブを生成するように構成され、前記事後処理手段は、前記複合ジョブを受任したノードにおいて、複合ジョブに含まれる各ジョブのうち当該ノードが実行すべきジョブの実行後、所定の場合に残余のジョブを他のノードに依頼するように構成されたことを特徴とする請求項1記載の分散処理システム。

【請求項5】 前記補助情報が、非実行形態の命令を実行形態の命令に変換するための手順を示す情報であることを特徴とする請求項1記載の分散処理システム。

【請求項6】 前記補助情報が、スクリプトの実行環境を表す情報であることを特徴とする請求項1記載の分散処理システム。

【請求項7】 通信媒体によって接続された複数のノードを含む分散処理システムにおいて、各ノードが、前記通信媒体を通じて他のノードと通信するための通信のステップと、

他のノードに依頼すべきジョブを生成する生成のステップと、

前記ジョブに、ジョブの命令の実行のための補助情報を付加する付加のステップと、

前記ジョブを前記通信のステップを通じて他のノードに送信して実行を依頼する依頼のステップと、

他のノードから送信されたジョブを受任する受任のステップと、

受任したジョブの前記補助情報に基づいて、当該ジョブの実行の準備を行う準備のステップと、

ジョブの命令を実行する実行のステップと、

前記実行の結果を処理する事後処理のステップと、を行うことを特徴とする分散処理方法。

【請求項8】 前記受任のステップでは、他のノードから送信されてきたジョブを受任するか否かを決定することを特徴とする請求項7記載の分散処理方法。

【請求項9】 前記依頼のステップでは、前記ジョブを全部又は一部の前記ノードに送信し、前記受任のステップでは、ジョブが送信されたノードにおいて、送信されたジョブを受任できると決定したときは、当該ジョブを送信した依頼側ノードに仮承諾を返信し、

前記依頼のステップでは、依頼側ノードにおいて、最早の前記仮承諾を返信したノードに正式依頼を送信することによってジョブの依頼を行うことを特徴とする請求項8記載の分散処理方法。

【請求項10】 前記生成のステップでは、所定の場合に複数のジョブからなる複合ジョブを生成し、前記事後処理のステップでは、前記複合ジョブを受任したノードにおいて、複合ジョブに含まれる各ジョブのうち当該ノードが実行すべきジョブの実行後、所定の場合に残余のジョブを他のノードに依頼することを特徴とする請求項7記載の分散処理方法。

【請求項11】 前記補助情報が、非実行形態の命令を実行形態の命令に変換するための手順を示す情報であることを特徴とする請求項7記載の分散処理方法。

【請求項12】 前記補助情報が、スクリプトの実行環境を表す情報であることを特徴とする請求項7記載の分散処理方法。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 本発明は、複数のノードが通信媒体を介して接続された分散処理システム及び分散処理方法の改良に関するものである。

【0002】

【従来の技術】 近年、複数のノード（コンピュータ）が通信媒体を介して接続された分散処理システム（ネットワーク）が広く普及しつつある。このような分散処理システムでは、ノード間でジョブを依頼しあって各種資源の有効活用を図っている。従来、ノード間でのジョブの

依頼では、依頼側ノードは被依頼側ノードに対してジョブの内容をなす命令と命令の実行において用いるデータを送信して依頼し、受任した側のノードは、ジョブの命令を実行し結果を依頼側ノードに返送していた（分散処理方法）。

【0003】このような従来の分散処理システム及び分散処理方法では、ジョブは依頼側から一方的に依頼されていた。また、複数のジョブを他のノードに連続的に依頼する場合、依頼はジョブごとに行われ、一つのジョブの実行が済むごとに実行結果が一旦依頼元のノードに返され、依頼元のノードはこの結果を受領してから後続するジョブを順次他のノードに依頼していた。

【0004】

【発明が解決しようとする課題】しかしながら、従来の分散処理システム及び分散処理方法では、被依頼側ノードは、依頼されるジョブの実行に必要な情報を有している必要があった。例えば、ジョブが、特定の高級言語によって記述されたソースコードの場合、このジョブを実行するノードは、当該言語のコンパイラを有していなければならない。このため、従来では、被依頼側ノードが必要な情報を有しているかを依頼側ノードが予めデータとして保持しておくか、その都度確認するという複雑な構成が必要であった。

【0005】また、どのノードにジョブを依頼するかを依頼側のノードが決定していたため、依頼が一部のノードに集中し、結果応答までの応答性を著しく損なうといった問題も発生する。さらに、従来では、連続する複数のジョブでもそれぞれ別個のノードに、結果返信を待ってから依頼していたため、通信媒体上のボトルネックが発生しやすく、複雑なジョブの状態管理が必要であった。

【0006】本発明は、上記のような従来技術の問題点を解決するために提案されたもので、その目的は、構成が単純な分散処理システム及び分散処理方法を提供することである。また、本発明の他の目的は、処理効率の優れた分散処理システム及び分散処理方法を提供することである。

【0007】

【課題を解決するための手段】上記の目的を達成するため、請求項1の分散処理システムは、通信媒体によって接続された複数のノードを含む分散処理システムにおいて、各ノードが、前記通信媒体を通じて他のノードと通信するための通信手段と、他のノードに依頼すべきジョブを生成する生成手段と、前記ジョブに、ジョブの命令の実行のための補助情報を付加する付加手段と、前記ジョブを前記通信手段を通じて他のノードに送信して実行を依頼する依頼手段と、他のノードから送信されたジョブを受任する受任手段と、受任したジョブの前記補助情報に基づいて、当該ジョブの実行の準備を行う準備手段と、ジョブの命令を実行する実行手段と、前記実行の結

果を処理する事後処理手段と、を有することを特徴とする。

【0008】また、請求項2の発明は、請求項1記載の分散処理システムにおいて、前記受任手段は、他のノードから送信されてきたジョブを受任するか否かを決定するように構成されたことを特徴とする。

【0009】また、請求項3の発明は、請求項2記載の分散処理システムにおいて、前記依頼手段は、前記ジョブを全部又は一部の前記ノードに送信するように構成され、前記受任手段は、ジョブが送信されたノードにおいて、送信されたジョブを受任できると決定したときは、当該ジョブを送信した依頼側ノードに仮承諾を返信するように構成され、前記依頼手段は、依頼側ノードにおいて、最早の前記仮承諾を返信したノードに正式依頼を送信することによってジョブの依頼を行うように構成されたことを特徴とする。

【0010】また、請求項4の発明は、請求項1記載の分散処理システムにおいて、前記生成手段は、所定の場合に複数のジョブからなる複合ジョブを生成するように構成され、前記事後処理手段は、前記複合ジョブを受任したノードにおいて、複合ジョブに含まれる各ジョブのうち当該ノードが実行すべきジョブの実行後、所定の場合に残余のジョブを他のノードに依頼するように構成されたことを特徴とする。

【0011】また、請求項5の発明は、請求項1記載の分散処理システムにおいて、前記補助情報が、非実行形態の命令を実行形態の命令に変換するための手順を示す情報であることを特徴とする。

【0012】また、請求項6の発明は、請求項1記載の分散処理システムにおいて、前記補助情報が、スクリプトの実行環境を表す情報であることを特徴とする。

【0013】また、請求項7の分散処理方法は、通信媒体によって接続された複数のノードを含む分散処理システムにおいて、各ノードが、前記通信媒体を通じて他のノードと通信するための通信のステップと、他のノードに依頼すべきジョブを生成する生成のステップと、前記ジョブに、ジョブの命令の実行のための補助情報を付加する付加のステップと、前記ジョブを前記通信のステップを通じて他のノードに送信して実行を依頼する依頼のステップと、他のノードから送信されたジョブを受任する受任のステップと、受任したジョブの前記補助情報に基づいて、当該ジョブの実行の準備を行う準備のステップと、ジョブの命令を実行する実行のステップと、前記実行の結果を処理する事後処理のステップと、を行うことを特徴とする。

【0014】また、請求項8の発明は、請求項7記載の分散処理方法において、前記受任のステップでは、他のノードから送信されてきたジョブを受任するか否かを決定することを特徴とする。

【0015】また、請求項9の発明は、請求項8記載の

分散処理方法において、前記依頼のステップでは、前記ジョブを全部又は一部の前記ノードに送信し、前記受任のステップでは、ジョブが送信されたノードにおいて、送信されたジョブを受任できると決定したときは、当該ジョブを送信した依頼側ノードに仮承諾を返信し、前記依頼のステップでは、依頼側ノードにおいて、最早の前記仮承諾を返信したノードに正式依頼を送信することによってジョブの依頼を行うことを特徴とする。

【0016】また、請求項10の発明は、請求項7記載の分散処理方法において、前記生成のステップでは、所定の場合に複数のジョブからなる複合ジョブを生成し、前記事後処理のステップでは、前記複合ジョブを受任したノードにおいて、複合ジョブに含まれる各ジョブのうち当該ノードが実行すべきジョブの実行後、所定の場合に残余のジョブを他のノードに依頼することを特徴とする。

【0017】また、請求項11の発明は、請求項7記載の分散処理方法において、前記補助情報が、非実行形態の命令を実行形態の命令に変換するための手順を示す情報であることを特徴とする。

【0018】また、請求項12の発明は、請求項7記載の分散処理方法において、前記補助情報が、スクリプトの実行環境を表す情報であることを特徴とする。

【0019】

【作用】上記のような構成を有する本発明は、次のような作用を有する。すなわち、請求項1、7の分散処理システムでは、あるノードで、他のノードに依頼すべきジョブが生成される。ジョブは、実行すべき手続きと実行に用いるデータを含む。ジョブには、ジョブの手続きの実行のための補助情報が付加される。

【0020】ジョブは通信媒体を通じて他のノードに送られる。ジョブを受任したノードでは補助情報に基づいてジョブ実行の準備が行われ命令が実行される。実行結果はジョブで指示されたように扱われるが、典型的には要求先に返送される。

【0021】このように請求項1、7の発明によれば、ジョブとともに補助情報が送信されるので、各ノードが処理可能なジョブの種類を依頼側で判断することが不要になり、構成が単純化される。

【0022】また、請求項2、8の発明では、被依頼側ノードが依頼されたジョブを受任するか否かを、自ノードの負荷などに基づいて決定する。このため、一部のノードへのジョブの集中が解決し、全体の処理効率が向上する。

【0023】また、請求項3、9の発明では、被依頼側ノードのうちジョブを受任できるものが仮承諾を依頼側ノードに返信し、依頼側ノードは最早の仮承諾を返信した被依頼側ノードに正式依頼を送信することによってジョブの依頼を行う。このような請求項3、9の発明によれば、各ノードでの処理が異なることがないので、構成

が単純化できる。

【0024】また、請求項4、10の発明では、実行すべき処理が複数のジョブから成る場合には、受任したノードから他のノードへ改めて依頼される。このため、依頼側ノードが各ジョブごとに依頼の処理を繰り返す必要がなくなり、処理効率が向上する。

【0025】また、請求項5、11の発明では、前記補助情報が、非実行形態の命令を実行形態の命令に変換するための手順を示す情報であるので、ジョブの手続き（命令）が実行形態であるか非実行形態であるかを問わず他のいずれのノードに依頼することもできる。このため、全体の処理効率が向上する。また、被依頼側がこのような手順を有するか否かの有無を依頼側で確認する必要がないので、構成が単純となる。

【0026】また、請求項6、12の発明では、前記補助情報が、スクリプトの実行環境を表す情報であるので、スクリプトの実行環境が被依頼側に存在するか否かを依頼側で確認する必要がなく、構成が単純化される。

【0027】

【実施例】次に、本発明の実施例について図面に従って具体的に説明する。なお、後述する実施例はコンピュータ上に実現され、実施例の各機能は所定の手順（プログラム）がこのコンピュータを制御することで実現される。

【0028】本明細書における各「手段」ないし「部」は実施例の各機能に対応する概念で、必ずしも特定のハードウェアやソフトウェア・ルーチンに1対1には対応しない。同一のハードウェア要素が場合によって異なった手段を構成する。例えば、コンピュータはある命令を実行するときにある手段となり別の命令を実行するときには別の手段となりうる。また、一つの手段がわずかに1命令によって実現される場合もあれば多数の命令によって実現される場合もある。

【0029】したがって、本明細書では、以下、実施例の各機能を有する仮想的回路ブロック（手段）を想定して実施例を説明する。但し、コンピュータの使用は一例であり、本発明の機能の全部又は一部は、可能ならば、カスタムチップ（専用の集積回路）のような電子回路上に実現してもよい。

【0030】実施例に用いられるコンピュータは、一般には、CPU（中央演算処理装置）と、RAM（随時書込読出型記憶素子）からなる主記憶装置とを有する。また、前記コンピュータの規模は自由であり、パーソナルコンピュータ・スモールコンピュータ・ワークステーション・メインフレームなど、いかなる規模のものを用いてもよい。

【0031】また、前記コンピュータのハードウェアは、典型的には、キーボードやマウスなどの入力装置と、ハードディスク装置などの外部記憶装置と、CRT表示装置やプリンタ印字装置などの出力装置と、必要な

入出力制御回路を含む。

【0032】但し、前記コンピュータのハードウェア構成は自由であり、本発明が実施できる限り、上記の構成要素の一部を追加・変更・除外してもよい。例えば、CPUの種類は自由であり、CPUを複数同時に用いたり、単一のCPUをタイムシェアリング（時分割）で使い、複数の処理を同時平行的に行ってもよい。

【0033】また、他の入力装置（例えば、タッチパネル・ライトペン・トラックボールなどのポインティングデバイスや、デジタイザ・イメージ読取装置やビデオカメラなどの画像入力装置・音声識別装置・各種センサなど）を用いてもよい。また、他の外部記憶装置（例えば、フロッピーディスク装置・RAMカード装置・磁気テープ装置・光学ディスク装置・光磁気ディスク装置・バブルメモリ装置・フラッシュメモリなど）を用いてもよい。また、他の出力装置（例えば、液晶表示装置・プラズマディスプレイ装置・ビデオプロジェクター・LED表示装置・音響発生回路・音声合成回路など）を用いてもよい。

【0034】また、前記コンピュータにおいて実施例を実現するためのソフトウェアの構成としては、典型的には、実施例の各機能を実現するためのアプリケーションプログラムが、OS（オペレーティングシステム）上で実行される態様が考えられる。また、実施例を実現するためのプログラムの態様としては、典型的には、高級言語やアセンブラからコンパイル（翻訳）された機械語が考えられる。但し、前記コンピュータのソフトウェア構成も自由であり、本発明が実施できる限り、ソフトウェア構成を変更してもよい。例えば、必ずしもOSを用いる必要はなく、また、プログラムの表現形式も自由であり、BASICのようなインタプリタ（逐次解釈実行型）言語を用いてもよい。

【0035】また、プログラムの格納態様も自由であり、ROM（読出し専用メモリ）に格納しておいてもよく、また、ハードディスク装置のような外部記憶装置に格納しておき、コンピュータの起動時や処理の開始時に主メモリ上にロード（読み込み）してもよい。また、プログラムを複数の部分に分割して外部記憶装置に格納しておき、処理内容に応じて必要なモジュールのみを随時主メモリ上にロード（読み込み）してもよい。さらに、プログラムの部分ごとに異なった態様で格納してもよい。

【0036】また、本実施例における各手順の各ステップは、その性質に反しない限り、実行順序を変更し、複数同時に実行し、また、実行ごとに異なった順序で実行してもよい。このような順序の変更は、例えば、ユーザが実行可能な処理を選択するなどメニュー形式のインターフェース手法によって実現することができる。

【0037】また、本明細書における「入力」は、本来の情報の入力のみならず、情報の入力と密接に関連する

他の処理を含む。このような処理は、例えば、入力内容のエコーバックや修正・編集である。また、「入力」は、必ずしもコンピュータ外部からの入力には限定されず、所定のメモリ領域又はディスク装置から情報を読み出すことによって行われる。また、本明細書における「出力」は、本来の情報の出力のみならず、情報の出力と密接に関連する他の処理を含む。このような処理は、例えば、出力すべき範囲の入力や、画面スクロールの指示である。また、「出力」は、必ずしもコンピュータ外部への出力には限定されず、所定のメモリ領域又はディスク装置に情報を書き込むことによって行われる。なお、対話的入出力手順によって入力と出力を一体的操作によって実現してもよく、このような一体的操作によって、選択・指定・特定などの処理を行ってもよい。

【0038】また、本明細書におけるデータ（情報）やデータの格納手段は前記コンピュータ上においていかなる態様で存在してもよい。例えば、データのハードウェア上の所在部分は、主記憶装置・外部記憶装置・CPUのレジスタやキャッシュメモリなどいかなる部分でもよい。また、データの保持態様も自由である。例えば、データは、ファイル形式で保持されるのみならず、メモリやディスクなどの記憶装置を物理的アドレスで直接アクセスすることによって実現してもよい。また、データの表現形式も自由で、例えば、文字列を表すコードの単位は、文字単位でも単語単位でもよい。また、データは必要とされる一定時間だけ保持されれば十分で、その後消滅してもよく、保持時間の長短は自由である。また、辞書データのように当面変更されない情報は、ROMに格納してもよい。

【0039】また、本明細書において、特定の情報への言及は確認的で、言及されない情報の存在を否定するものではない。すなわち、本発明の動作では、動作に必要な一般的な情報やその格納領域、例えば、各種ポインタ、スタック、カウンタ、フラグ、パラメータ、ワークエリア、バッファなどが適宜用いられる。

【0040】実施例の各部分が処理に要する情報は、特に記載がない場合、当該情報を保持している他の部分から獲得される。このような情報の獲得は、例えば、当該情報を格納している変数やメモリをアクセスすることによって実現することができる。なお、情報の消去・抹消は、当該情報の内容自体を必ずしも記憶領域から現実には削除せず、消去を表すフラグを設定するなど、情報の意味付けの変更によって行うことができる。

【0041】（1）実施例の構成

本実施例は、通信媒体によって接続された複数のノードを含む分散処理システム及びこの分散処理システムにおける分散処理方法であり、請求項1～12に相当する。本実施例の目的は、構成が単純な分散処理システム及び分散処理方法を提供することである。また、本実施例の他の目的は、処理効率の優れた分散処理システム及び分

散処理方法を提供することである。

【0042】まず、図1は本実施例の分散処理システムにおいて通信媒体1に接続されたノードの一つについて、その構成を示す機能ブロック図である。各ノードは、図1に示すように、他のノードに依頼すべきジョブ2を生成する生成手段3と、前記ジョブ2に、ジョブ2の手続きの実行のための補助情報を付加する付加手段4と、ジョブ2を通信手段5及び通信媒体1を通じて他のノードに送信して実行を依頼する依頼手段6と、を有する。また、各ノードは、他のノードからジョブ2を受任する受任手段7と、受任したジョブ2の前記補助情報に基づいて、当該ジョブ2を実行するための準備を行う準備手段8と、ジョブの命令を実行する実行手段9と、前記実行の結果を処理する事後処理手段10と、を有する。

【0043】(2)実施例の作用及び効果

上記のような構成を有する本実施例は次のような作用を有する。

【0044】〔概略的作用〕図2は、本実施例におけるジョブの依頼とジョブの実行の手順の一例を示すフローチャートである。すなわち、本実施例の分散処理システム及び分散処理方法では、あるノードで、他のノードに依頼すべきジョブ2が生成される(ステップ21)。ジョブ2は、実行すべき手続きと実行に用いるデータを含む。ジョブ2には、ジョブ2の手続きの実行のための補助情報が付加される(ステップ22)。ジョブ2は通信手段5及び通信媒体1を通じて他のノードに依頼される(ステップ23)。ジョブ2を受任したノードでは(ステップ24)、補助情報に基づいてジョブ2実行の準備が行われる(ステップ25)。この準備は、例えば非実行形態の手続きを実行形態に変換したり、所定の環境の設定などである。準備の後、ジョブ2に含まれる手続き(命令)が実行される(ステップ26)。実行結果はジョブ2で指示されたように扱われるが(ステップ27)、典型的には要求先(依頼元)に返送・受信される(ステップ87)。

【0045】〔ジョブの生成〕まず、各ノードにおいて、生成手段3が他のノードに依頼すべきジョブ2を生成する。依頼すべきジョブ2は、明示的に実行依頼を示したアプリケーション・プログラム又はスケジューラの動作に基づいて内容や属性が決定される。

【0046】ここで、図3はジョブ2の生成の手順を示すフローチャート、図4は生成されるジョブ2ごとのデータ構造を示す。すなわち、各ジョブ2はこの図4に示すようにヘッダ部、手続き部、データ部から構成される。このうちヘッダ部はジョブ2全体の属性を示す形式的情報、手続き部はジョブ2の内容として実行されるべき命令群、データ部は手続きの実行に要するデータであり、必要な場合にセットされる。

【0047】ここで、図5は、ヘッダ部のデータ構造を

示す図である。ジョブ2の生成では、まず、このうち「手続きの属性」が決定される。「手続きの属性」とは、手続きがロードモジュールのような実行形態であるか、ソースプログラムやスクリプトのような非実行形態であるかといった種類と、処理に必要な情報が付属しているか、また、処理に必要な情報が直前の手続きの結果を使用するかどうかといった各種種別を示し、実際には所定のフラグをセットすることにより表現される(図3)。

【0048】例えば、実行形態(ロードモジュール)の場合はExe(ステップ31, 32)、ソースプログラムの場合はSrc(ステップ34, 35)、スクリプトの場合はCmdなどのフラグがセットされる(ステップ38, 39)。また、ジョブ2に何らかの情報が付属している場合は(ステップ42) data(ステップ43)、ジョブ2の処理に直前の手続きの結果を情報として使用する場合は(ステップ44) cont(ステップ45)などのフラグがセットされる。

【0049】なお、付加手段4は、ジョブ2の手続きの実行のための補助情報をジョブ2に付加する。例えば、ジョブ2の命令が非実行形態の場合はこの命令を実行形態に変換するための手順がセットされる。具体的には、命令がソースプログラムの場合はコンパイル・リンクの手順を示したコマンド列(コンパイラ・リンク)すなわちロードモジュールの生成手順をセットし(ステップ37)、命令がスクリプトの場合は実行環境を表すキーワードなどの情報(例えばcsh)をセットする(ステップ41)。

【0050】次に、ヘッダ中の「制約条件」が決定される(図5)。制約条件は手続きを実行するための制約事項であり主にアクセスコントロールに関して実行ユーザID、実行ディレクトリを定義する。この内容となる値は、必要な場合にはアプリケーション又はスケジューラによって与えられる。

【0051】さらに、ヘッダ最後の「実行後の処理」がセットされる。通常、手続きが実行された結果を要求元に返送する場合にReturn(ステップ46, 47)、他のノードに転送する場合はSend(ステップ48, 49)がセットされる。「実行後の処理」は、一定の目的を達成するため複数連続したものとして生成されるジョブ2(以下「複合ジョブ」という)の場合には次のようにセットされる。なお、図6は、複合ジョブのデータ構造を示す。複合ジョブを構成する各ジョブ2は上記のような手順の繰り返しによって生成される。

【0052】複合ジョブの場合、次に継続手続きがあり次の手続きの結果の転送を待つ場合はNextWaitを(ステップ50, 51, 53)、手続きの処理結果を次の手続きに引き渡す場合はNextDataをセットする(ステップ51, 52)。なお、Sendの場合のみ、その直後に転送先ネットワークアドレスを定義する(ス

テップ49)。この項目が設定されておらずデフォルトの場合、受任側ノードでは命令実行後何もせず終了となる。

【0053】なお、ジョブ2には、ジョブ2をネットワーク上で一意に識別するためのジョブIDが生成・付加される。ジョブIDは図7に示すようにネットワーク上の位置を示すネットワークアドレス、要求ユーザのユーザID、要求コンピュータ内での要求シリアル番号の組み合わせを含む。図8は、ジョブIDが付加されたジョブ2の状態を示す。

【0054】[ジョブの依頼と受任] ジョブ2が生成されると、依頼手段6が、ジョブ2を前記通信手段5を通じて他のノードに送信して実行を依頼する。依頼は、他のノードの受任手段7によって次のように処理され、ジョブ2の受任が成立する。ここで、依頼手段6は通信手段5及び通信媒体1を通じてジョブIDとジョブ2をブロードキャスト（同報通信）によって他の全てのノードに送信し、応答を待つ。図9は、依頼側ノードと被依頼側ノードとの間での通信の状態を示す。依頼手段6は、依頼する各ジョブ2の状態をテーブルで管理する。図10はジョブ2を管理するテーブルの構造の例を示す。

【0055】依頼側ノードが送信したジョブ2は通信媒体1及び通信手段5を通じて他のノードの受任手段7に到達する。このとき、被依頼側ノードの受任手段7は、他のノードから送信されてきたジョブ2を受任するか否かを決定する。図11は被依頼側ノードにおける受任の処理手順を示すフローチャートである。すなわち、被依頼側ノードの受任手段7はジョブ2が自身のノード上で実行可能か否かを判断する。

【0056】この判断は、まず、ジョブ2自体の種別・内容に基づいて行う。すなわち、実行形態のロードモジュールであれば（ステップ115）手続きが（ステップ116）コードセットなどとの関係で実行可能か（ステップ117）、非実行形態のソースプログラムであれば（ステップ118）メモリ容量等との関係において、補助情報たるコンパイラで実行モジュールが生成可能か（ステップ119）、スクリプトであれば（ステップ120）対応する実行環境が提供できるか（ステップ121）、によって判断される。ジョブ2自体が実行可能な場合は、自身のノードの負荷から当該ジョブ2を実行可能かが判断される（ステップ122）。なお、複数手続きで構成されるジョブ2の場合は先頭の手続きに関して上記のように判断する。

【0057】被依頼側ノードの受任手段7は、ジョブ2を実行できる場合は、仮承諾を依頼側ノードに返送する。ここで、図11のうちステップ113、114は便宜上、依頼側ノードの依頼手段6の動作を示したものである。すなわち、依頼側ノードの依頼手段6は、最早の仮承諾（最初の応答）を返送してきたノードに（ステップ113）正式依頼を送信するとともに、前記ジョブ管

理テーブルにジョブID、要求ユーザ、依頼日時、依頼先ノードを登録する（ステップ114）。

【0058】なお、依頼手段6は2番目以降に返送された仮承諾は無視し、一方、無視された仮承諾を発送した被依頼側ノードの受任手段7は、一定時間のタイムアウトによりこれを判定し、対応する要求依頼を消去する。

【0059】このように、本実施例では、被依頼側ノードが依頼されたジョブ2を受任するか否かを、自ノードの負荷などに基づいて決定する。このため、一部のノードへのジョブ2の集中が解決し、システム全体の処理効率が向上する。

【0060】また、本実施例では、上記のように、被依頼側ノードのうちジョブ2を受任できるものが仮承諾を依頼側ノードに返信し、依頼側ノードは最早の仮承諾を返信した被依頼側ノードに正式依頼を送信することによってジョブ2の依頼を行う。このような本実施例によれば、各ノードでの処理が異なることがなく、各ノードを同一の構成・手順で実現すればよいので、システム全体の構成が単純化できる。

【0061】[ジョブ実行の準備] 正式依頼の送信によって依頼が成立すると、依頼側ノードでは、まず、準備手段8が、受任したジョブ2の補助情報に基づいて、当該ジョブ2実行の準備を行う。すなわち、図12は、受任したノードにおける準備の手順を示すフローチャートである。準備手段8は、受任手段7から制御を引き渡された場合、該当するジョブ2を取り出し（ステップ1202、1203）、ジョブ2をヘッダ部、手続き部、データ部に分解する（ステップ1204）。なお、複数のジョブ2がキュー（待ち列）を構成している場合や受任しているジョブ2が複合ジョブである場合が考えられるが、いずれの場合も準備の対象はキューや複合ジョブに含まれるジョブ2のうち、未処理の最初のジョブ2である。次に、ヘッダ部が解析される。

【0062】このとき、「手続き属性」がExeの場合はジョブの命令は実行形態（ロードモジュール）であるから（ステップ1205）、メモリ領域の確保のような一般的処理が行われる（図示せず）他は、特段の処理は行われない。手続き属性がSrcの場合はジョブの命令はソースプログラムのため（ステップ1206）セットされている補助情報たる生成手順に従ってコンパイルを行なう（ステップ1207）。また、手続き属性がCmdの場合はジョブの命令はスクリプトであるから（ステップ1208）、「生成手順」にセットされている実行環境を取り出し、スクリプトと合わせて実行形態とする（ステップ1209）。

【0063】このように、本実施例では、前記補助情報が、非実行形態の命令を実行形態の命令に変換するための手順を示す情報を含むので、実行形態であるか非実行形態であるかを問わず他のいずれのノードに依頼することもできる。このため、全体の処理効率が向上する。ま

た、被依頼側がこのような手順を有するか否かの有無を依頼側で確認する必要がないので、構成が単純となる。

【0064】また、本実施例では、前記補助情報がスクリプトの実行環境を表す情報を含むので、スクリプトの実行環境が被依頼側に存在するか否かを依頼側が確認する必要がなく、構成が単純化される。

【0065】なお、準備の最後に、ヘッダ中の残りの情報である「制約事項」と「実行後の処理」を取り出す（ステップ1210）。ここで、「手続きの属性」中のContフラグがセットされているかをチェックし、セットされている場合は直前の手続きの結果を当該手続きの処理に使用するためその終了を待つ。すなわち、実行しようとするジョブ2が複合ジョブ中の2番目以降のジョブ2の場合、直前のジョブ2の結果の利用が指定されていることが考えられる。この場合、直前のジョブ2は他のノードに依頼されている場合も考えられるし、同一のノードにおいて並行処理の他のタスクによって実行されていることも考えられる。この場合、直前のジョブ2の結果を待つため、実行形態の手続きとデータおよび取り出した「制約事項」と「実行後の処理」を待ち状態とする。Contがセットされていない場合は実行形態の手続きとデータを共に実行手段9に渡す。

【0066】【ジョブの実行】準備の後、実行手段9がジョブ2の命令を実行する（ステップ1211）。つまり、実行手段9は、準備手段8より渡された実行可能な手続きを、データをパラメータとして実行する。この時、準備手段8より渡されるヘッダ中の「制約条件」のチェックを行なう。ここで、図13はジョブ2の実行及び実行後の処理の手順を示すフローチャートである。

【0067】すなわち、対象とするジョブ2をメモリから取り出し（ステップ1301）ヘッダ中のフラグ「制約条件」にUserID、Dirが指定されている場合は（ステップ1302）それを実行UserIDおよび作業ディレクトリにセットする（ステップ1303）。また、指定がない場合は（ステップ1302）デフォルトとして（ステップ1304）、要求元ノードの要求ユーザのユーザIDをUserIDとし、そのユーザのホームディレクトリを作業ディレクトリとする（ステップ1303）。デフォルトチェックによって、UserID、Dirの指定がなく実行ノード上で要求元のユーザが登録されていないことが判明した場合は、実行要求を棄却し通信手段5を通じて依頼元ノードにジョブ2を返送する（ステップ1305）。

【0068】続いて、実行手段9が、命令をデータに基づいて実行する（ステップ1306）。このとき、実行形態およびSrc（ソースプログラム）の属性を持った手続きは既に実行形態になっているため直接実行し結果を保存する。また、スクリプトである命令は「生成手順」に指定された環境下でスクリプトのまま実行し結果を保存する。

【0069】このように本実施例によれば、ジョブ2とともに補助情報が送信されるので、各ノードが処理可能なジョブ2の種類を依頼側で判断することが不要になり、構成が単純化される。

【0070】【実行後の処理】実行後、事後処理手段9が実行の結果を処理する。すなわち、「実行後の処理」にNextDataがセットされている場合は（ステップ1307）、結果を待っている次のジョブ2が存在しており、次のジョブ2は、終了したジョブ2（手続き）の実行結果をパラメータとして実行する必要がある。このため、事後処理手段9は、実行結果にジョブIDを付加し準備手段8に渡す（ステップ1308）。この場合、実行結果を受領した準備手段8は、この結果を次のジョブ2に渡すべく、後続する待ち状態のジョブ2のデータに、実行手段9より渡された結果を追加する。そして、後続するジョブ2についての準備を行った後、再度実行手段9に制御を引き渡す（ステップ1309）。

【0071】複合ジョブは、このように準備手段8と実行手段9との間で相互に制御をやり取りすることによって順次実行される。但し、複合ジョブの途中で当該ノードで実行できないものが含まれている場合は、実行できないジョブ2又は実行できないジョブ2を含む残りのジョブ2は他のノードに再度依頼される。この時、ジョブIDは要求元で作成したジョブIDを使用する。

【0072】このように、本実施例では、実行すべき処理が複数のジョブ2から成る場合には、受任したノードから他のノードへ改めて依頼される。このため、依頼側ノードが各ジョブ2ごとに依頼の処理を繰り返す必要がなくなり、処理効率が向上する。

【0073】なお、「実行後の処理」にNextDataがセットされていないものについては次のように結果が送信される（ステップ1310）。すなわち、「実行後の処理」がReturn（要求元への返送）であれば、実行結果は通信手段5に渡され、依頼側ノードに返送される。また、「実行後の処理」がSend（指定先への転送）であれば、実行結果は通信手段5及び通信媒体1を通じて指定のノードに転送される。なお、図14は「実行後の処理」がReturnであるかSendであるかによって実行結果の送信先が異なる様子を示す概念図である。

【0074】【ジョブの処理の実例】上記のような本実施例におけるジョブ2の処理の実例を示す（図15）。例えば、ノードN1でデータD1を用いた手続きP1（ロードモジュール）と、その結果を用いる手続きP2（ソースプログラム）からなる複合ジョブJ1が生成されたと仮定する。この場合、手続きP1には補助情報としてコンパイラC1付加される。

【0075】被依頼側ノードN2、N3、N4がロードキャストされたジョブJ1を受信する。ノードN2はメモリ容量不足のため手続きP1が実行できないので仮

承諾は返信しない。ノードN3、N4は実行可能と判断し仮承諾をノード1に返信し、このうちノードN3の仮承諾がノードN1に先着した。ノードN1はノードN3に正式依頼を返信し、これによってノードN1からノードN3へのジョブJ1の依頼が成立する。

【0076】ノードN1はノードN4からの仮承諾を無視し、一方、ノードN4は正式依頼が返信されないのでタイムアウトによりジョブを消去する。ノードN3では、手続きP1はそのまま実行され、「実行後の処理」nextdataに基づいてその結果Rが手続きP2のデータD2に付加される。ソースプログラムである手続きP2はコンパイラCによって実行形態にコンパイルされ、データD2+Rに基づいて実行される。手続きP2の「実行後の処理」がreturnの場合、最終の実行結果はノードN1に返送される。

【0077】(3) 他の実施例

本発明は上記実施例に限定されるものではなく、実施態様の変更は自由であるから、次に例示するような他の実施例をも包含するものである。例えば、通信手段はブロードキャスト形式(同報通信)には限定されず、ノード間でジョブを順次転送するなど他の形式でもよい。また、補助情報はコンパイラやスクリプトの環境には限定されず、他の内容でもよい。また、受任手段は必ずしも自ら受任の可否を決定しなくともよく、この場合は、例えばスケジューラが各ノードの負荷に基づいて各ジョブの依頼先を振り分けてもよい。また、受任の手続きでは必ずしも仮承諾と正式依頼を用いる必要はなく、例えば最早に受任することを決定したノードが他の全てのノードに受任宣言を送信するなど他の手続きを用いてもよい。また、本発明の分散処理システム及び分散処理方法では、必ずしも複合ジョブを扱える必要はない。

【0078】

【発明の効果】以上説明したように本発明によれば、構成が単純な分散処理システム及び分散処理方法を提供することができるので、分散処理システム及び分散処理方法を構築する費用が削減され、システムのメンテナンスも容易になる。

【図面の簡単な説明】

【図1】本発明の実施例の分散処理システムにおいて通信媒体1に接続されたノードの一つについて、その構成を示す機能ブロック図。

【図2】本発明の実施例の分散処理システムにおけるジョブの依頼とジョブの実行の手順の一例を示すフローチャート。

【図3】本発明の実施例の分散処理システムにおけるジョブの生成の手順を示すフローチャート。

【図4】本発明の実施例の分散処理システムにおいて生成されるジョブごとのデータ構造を示す図。

【図5】本発明の実施例の分散処理システムにおけるジョブのヘッダ部のデータ構造を示す図。

【図6】本発明の実施例の分散処理システムにおける複合ジョブのデータ構造を示す図。

【図7】本発明の実施例の分散処理システムにおけるジョブIDのユーザID、要求コンピュータ内での要求シリアル番号の組み合わせを示す図。

【図8】本発明の実施例の分散処理システムにおいてジョブIDが付加されたジョブの状態を示す図。

【図9】本発明の実施例の分散処理システムにおける依頼側ノードと被依頼側ノードとの間での通信の状態を示す図。

【図10】本発明の実施例の分散処理システムにおけるジョブを管理するテーブルの構造の例を示す図。

【図11】本発明の実施例の分散処理システムにおける被依頼側ノードによるジョブの受任の処理手順を示すフローチャート。

【図12】本発明の実施例の分散処理システムにおける受任したノードでのジョブ実行の準備の手順を示すフローチャート。

【図13】本発明の実施例の分散処理システムにおけるジョブの実行及び実行後の処理の手順を示すフローチャート。

【図14】本発明の実施例の分散処理システムにおいて、「実行後の処理」によって実行結果の送信先が異なる様子を示す概念図。

【図15】本発明の実施例の分散処理システムにおけるジョブの処理の実例を示す図。

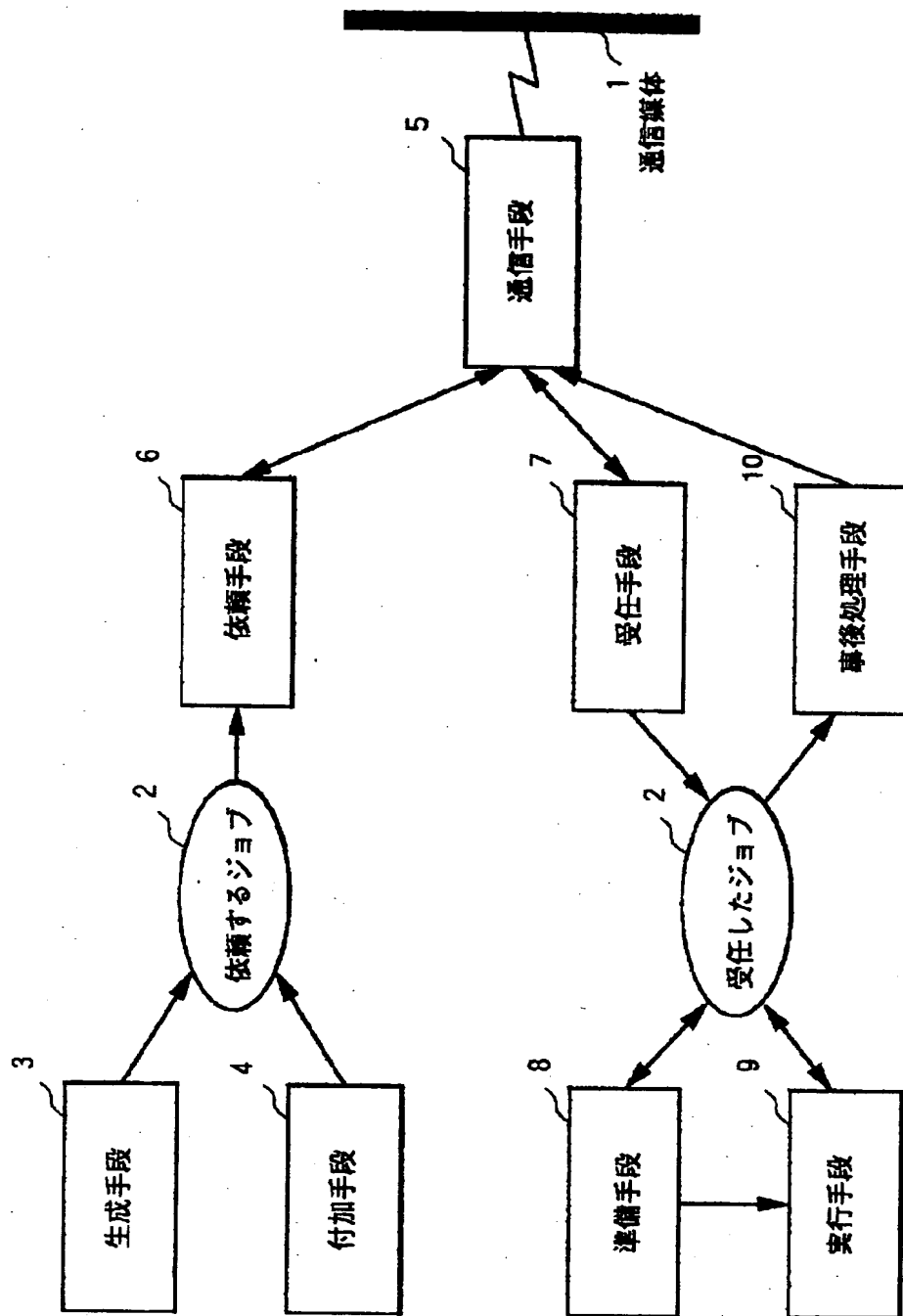
【符号の説明】

- 1：通信媒体
- 2：ジョブ
- 3：生成手段
- 4：付加手段
- 5：通信手段
- 6：依頼手段
- 7：受任手段
- 8：準備手段
- 9：実行手段
- 10：事後処理手段

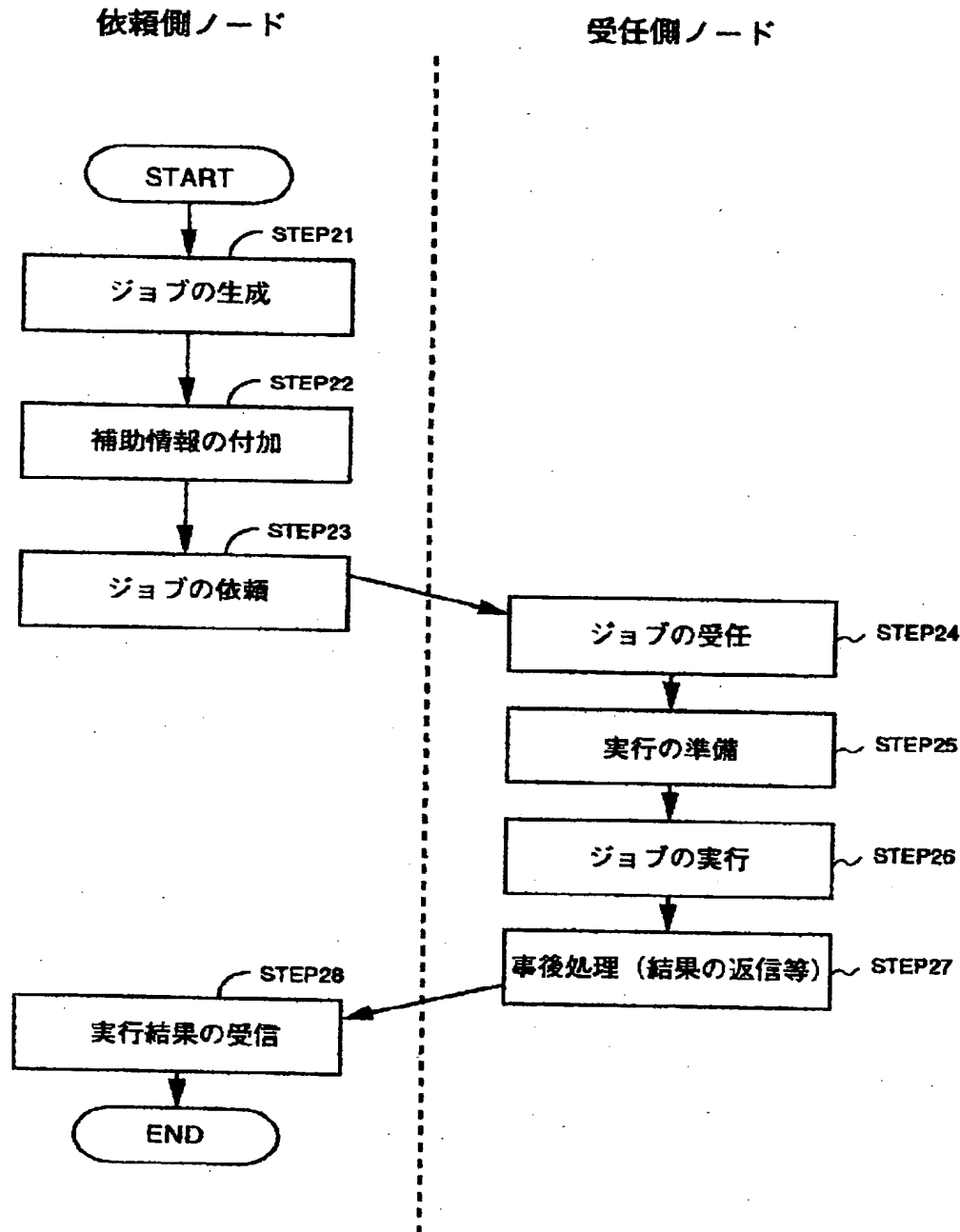
【図7】

| | | | |
|-------|----------------|-------|--------|
| ジョブID | ネットワーク アドレス | ユーザID | シリアル番号 |
|-------|----------------|-------|--------|

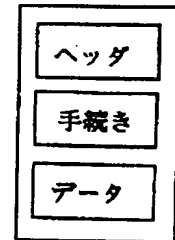
【図1】



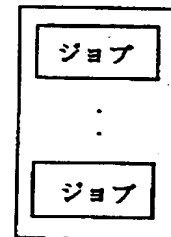
【図2】



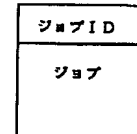
【図4】



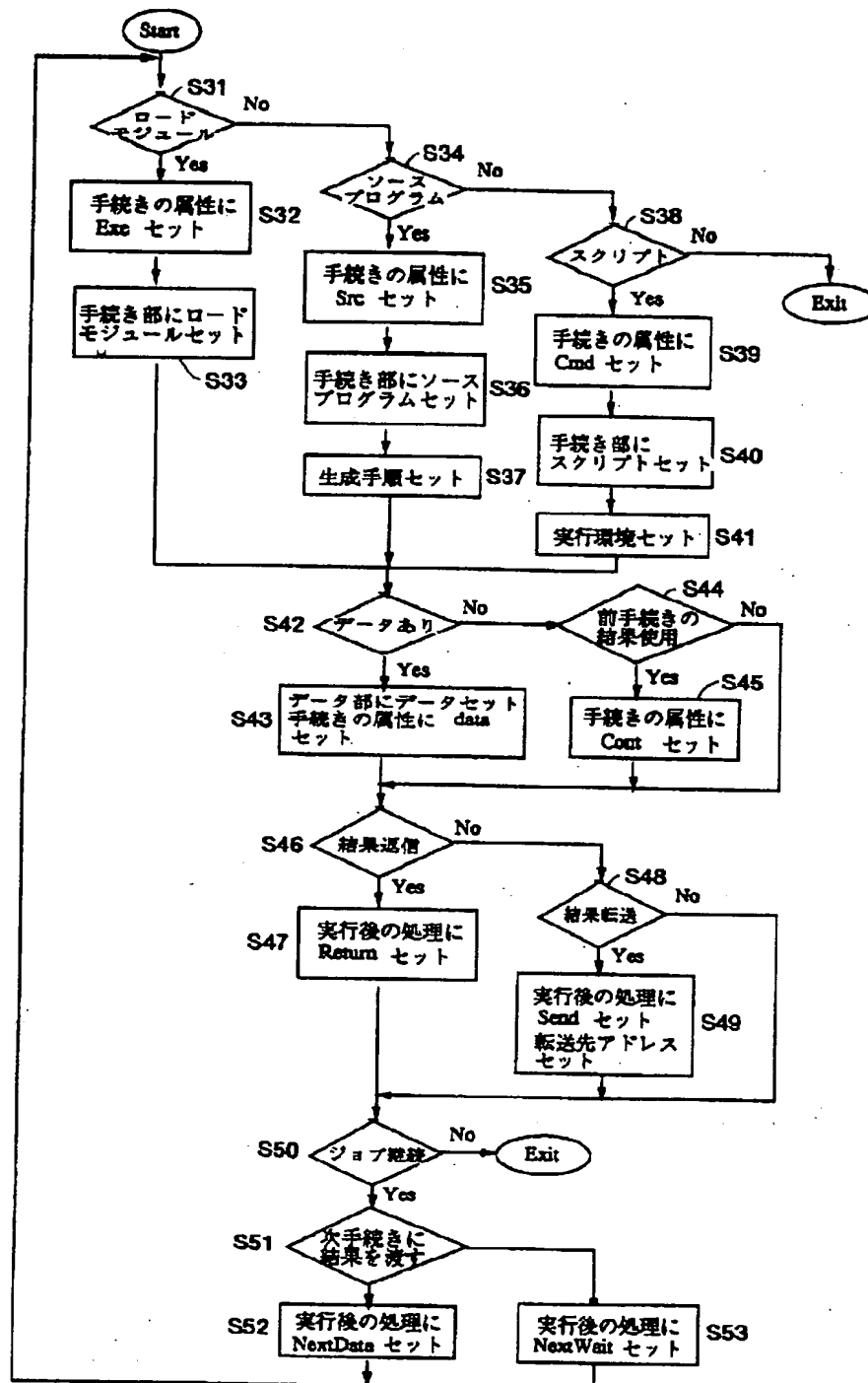
【図6】



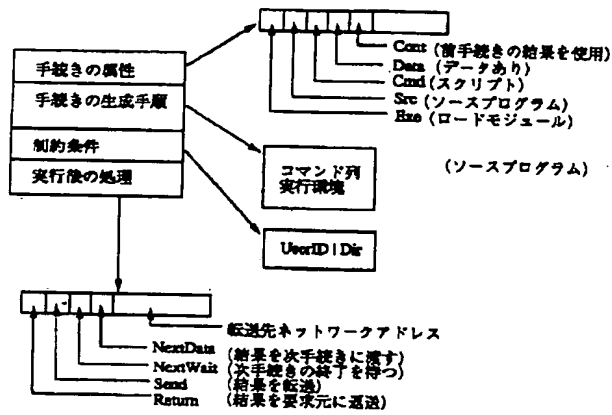
【図8】



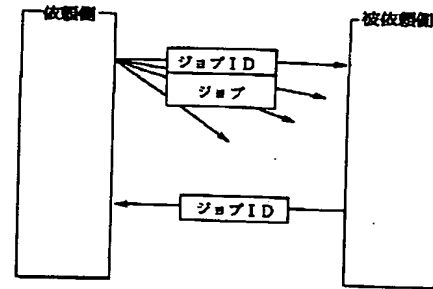
【図3】



【図5】



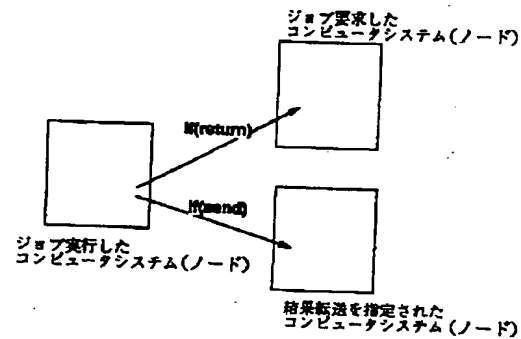
【図9】



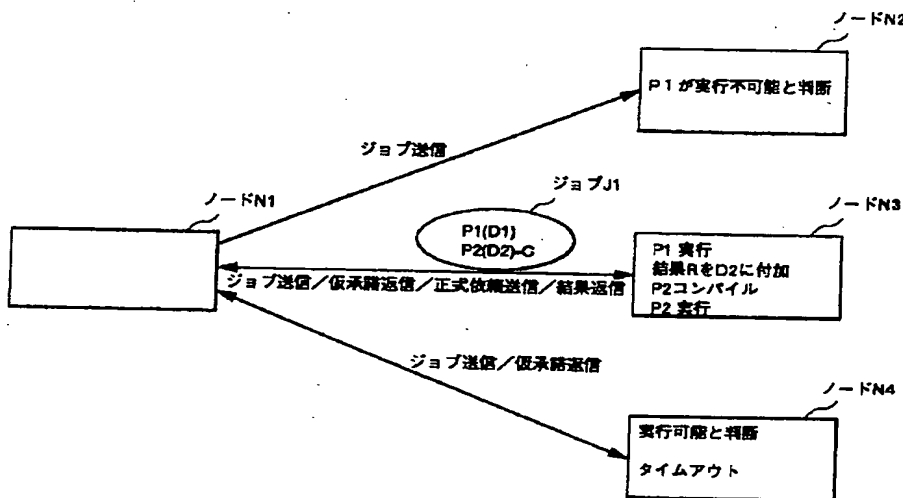
【図10】

| ジョブID | ネットワーク アドレス | ユーザID | 日時 |
|-------|----------------|-------|----|
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

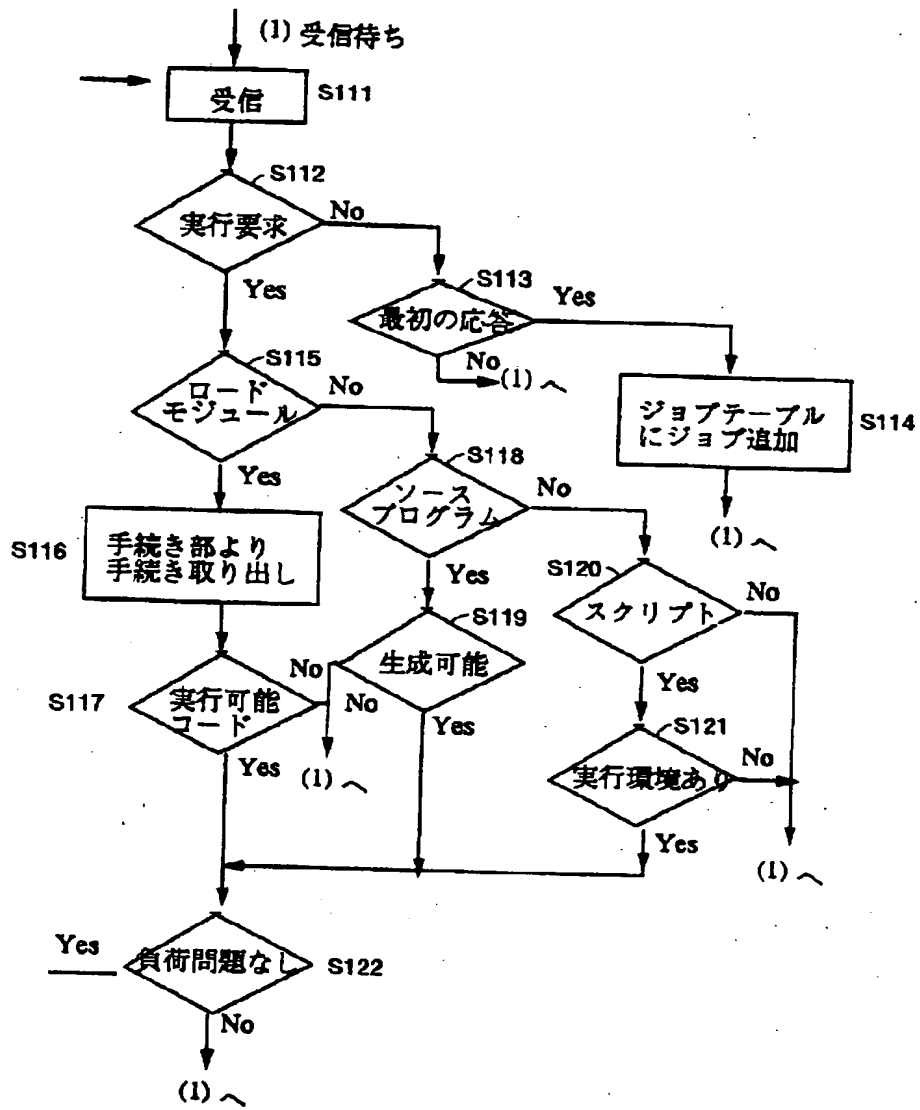
【図14】



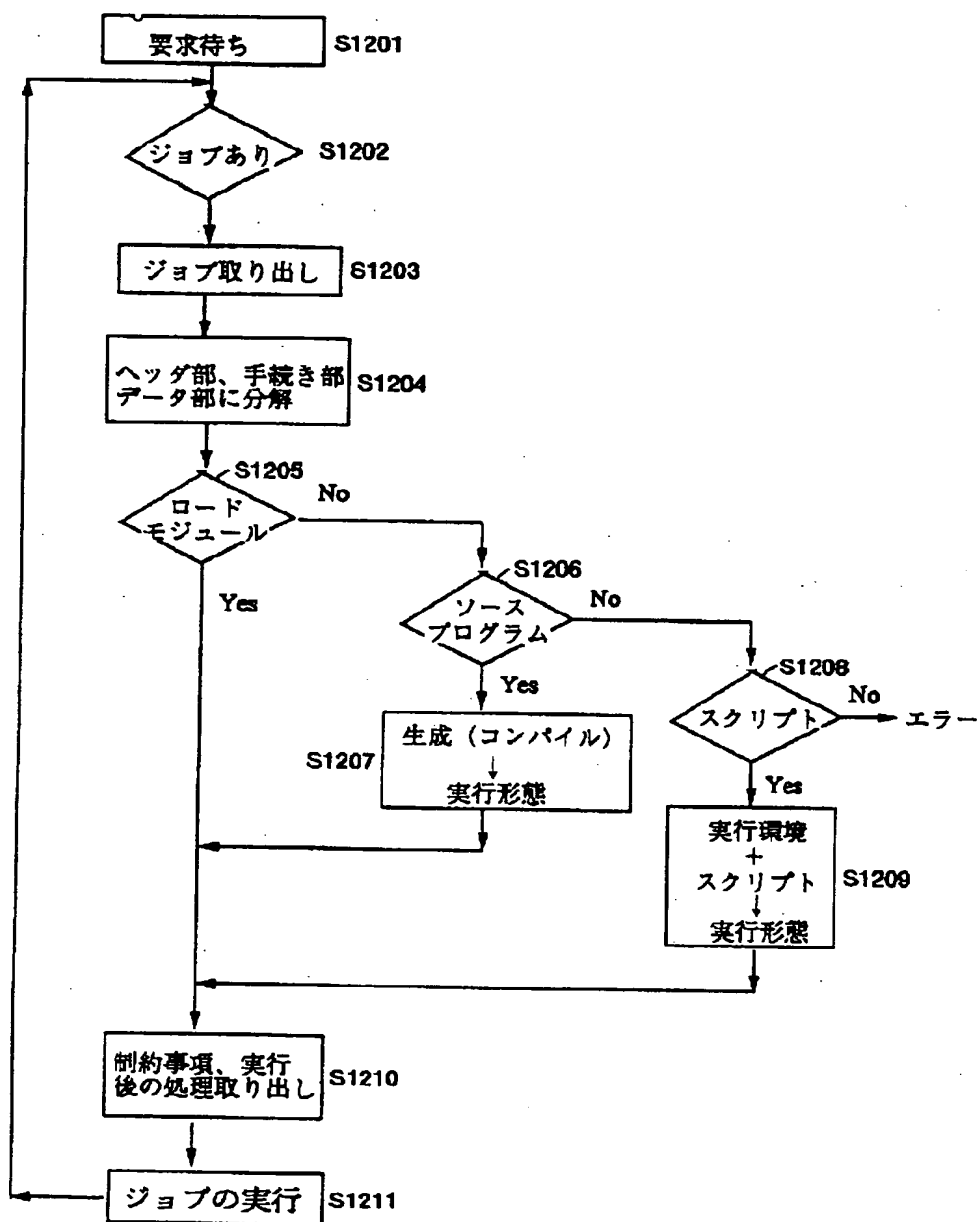
【図15】



【図11】



【図12】



【図13】

